Comp. & Maths. with Appls. Vol. 7, No. 5, pp. 431–438, 1981
Printed in Great Britain.
0097–4943/81/050431–08$02.00/0
Pergamon Press Ltd.

THE BIT-OPERATION COMPLEXITY OF MATRIX MULTIPLICATION AND OF ALL PAIR SHORTEST PATH PROBLEM

V. YA. PAN†

The Institute for Advanced Study, School of Mathematics, Princeton, NJ 08540, U.S.A.

Communicated by E. Y. Rodin

(*Received January* 1981)

Abstract—An $N \times N$ matrix product can be evaluated with precision $E > 0$ in $O(N^{s+\epsilon} \log (M/E) \log\log (M/E) \log\log\log (M/E))$ bit-operations, $\epsilon > 0$ arbitrary, M the maximum absolute value of the entries of given matrices, $O(n^s)$ the arithmetic complexity of $n \times n$ matrix multiplication, $s < 2.496$. The shortest path problem for a graph with N vertices whose edges have non-negative integer costs and all shortest distances are bounded by H can be solved in $O(N^{s+\epsilon}H)$ bit-operations.

1. INTRODUCTION

In recent years several asymptotically fast straight-line algorithms have been designed to perform $n \times n$ matrix multiplication (hereafter referred to as MM) involving $O(n^s)$ arithmetic operations, $s < 3$; see [1–9]. A further acceleration of MM is expected.

In the present paper we prove that all asymptotically fast arithmetic algorithms for MM (assuming that n grows and including also the algorithms that have not been designed yet) can be transformed into almost equally rapid algorithms for MM associated with bilinear decompositions whose constants are bounded by n^ν; ν does not depend on n. Such a bound is obtained by applying recursively a fixed basic algorithm for $\bar{N} \times \bar{N}$ matrix multiplication, $\bar{N}$ fixed. h many recursive applications define an algorithm that is called the h-th tensorial power of the basic algorithm and that multiplies $\bar{N}^h \times \bar{N}^h$ matrices. The above bound on the constants is general for bilinear decompositions associated with tensorial powers of a bilinear algorithm. The bound can be extended to the class of algorithms for polynomial multiplication and can be interpreted as a weak (exponential) stability of algorithms. (Compare with [10] where the very strong stability of fast algorithms for MM is denied.)

In this paper we use the weak stability to estimate an upper bound on the bit-operation complexity of an approximate solution of a given weakly stable problem with a given precision, E. The bound is

$$O(n^{s+\epsilon} \log (M/E) \log\log (M/E) \log\log\log (M/E)). \tag{1.1}$$

Here, $\epsilon > 0$ arbitrary, M is the maximum absolute value of input variables, $O(n^s)$ is the number of arithmetic operations sufficient for the exact solution of the problem. We prove that estimate for the MM problem where $s < 2.496$ ((1.1) can be extended to the multiplication of two n-th degree polynomials) by applying the original exactly computing algorithm, ECA, (we take the name from [3]) to the case where all values of variables and all constants of the ECA are also replaced by their approximations. All approximations are obtained by the truncations of the approximated values to sufficiently many binary digits. (The round off would also do.) We use the binary representation with fixed radix point.

In another paper [11], we show that the upper bounds of that kind are implied by the weak stability rather than by the mere existence of ECA's for the same problem. We do that by proving the lower bound $\Omega(n^2)$ on the bit-operation complexity of n-th degree polynomial multiplication over the class of evaluation-interpolation algorithms with real constants while arithmetic complexity of the same problem is $O(n \log^2 n)$; see ([12], pp. 101–102).

†This research has been supported in part by a grant from the National Science Foundation (Grant No. 800 3347).

We state our basic result in the next section and prove it in Sections 3–5. In Section 6 we outline a simpler proof that is based on another transformation of the original algorithm and can be applied if the constants of the basic straight-line arithmetic algorithm for MM are rational. (All existent fast algorithms for MM can be transformed into almost equally fast algorithms for MM with rational constants.) In Section 7 we combine our result with the algorithm due to [13]. This enables us to find the matrix of costs of all pair shortest paths of a graph with n vertices in $O(n^{s+\epsilon}H)$ bit-operations in the case where the given costs of edges are nonnegative integers and all shortest distances are bounded by H and where s, ϵ are as in (1.1). In particular, if $H=O(N^{3-s-\delta})$ for some $\delta>0$, then we choose $\epsilon<\delta$ and reduce the exponent 3 of the best previously known upper bound on the bit-complexity of the shortest path problem (due to [14]).

Several complexity measures for MM including the fixed precision complexity (which coincides with what we call the bit-operation complexity) are considered in [15]. However, the main direction of our study is different from [15] where only the upper bound $O(n^s \log n \log\log n \log\log\log n)$ is given without a proof. Such a bound, unlike (1.1), does not lead to the estimate $O(n^{s+\epsilon}H)$ on the bit-complexity of the shortest path problem.

2. THE BASIC THEOREM

Throughout this paper all logarithms are to the base 2, all numbers are assumed to be represented in binary form with fixed radix point, all norms of the matrices are in l_∞, that is, $\|G\|=\max\limits_{i,j}|g_{ij}|$ for a matrix $G=(g_{ij})$. If G, H are $N\times N$ matrices, then

$$\|GH\|\le N\|G\|\cdot\|H\|,\ \|G+H\|\le\|G\|+\|H\|. \tag{2.1}$$

THEOREM 2.1.

Given $\epsilon>0$, $M=M(N)$, $E=E(N)$ and two $N\times N$ matrices, $X=(x_{ij})$, $Y=(y_{ij})$, with real or complex entries, such that $M>1>E>0$.

$$\|X\|<M,\ \|Y\|<M, \tag{2.2}$$

then the matrix product $Z=XY$ can be evaluated with the precison $2E$ in $O(N^{s+\epsilon}\log(M/E)\log\log(M/E)\log\log\log(M/E))$ bit-operations assuming that $n\times n$ matrix multiplication can be performed by a straight-line algorithm involving $O(n^s)$ arithmetic operations.

Remark 2.1. We can assume (see [9]) that $2\le s<2.496$.

To simplify the proof of Theorem 2.1, we partition it into several steps and also assume that X, Y are real matrices. (The complex case is easily reduced to the real one. Indeed, if $X=T+iU$, $Y=V+iW$, then $XY=TV-UW+i(TW+UV)$ where i is the square root of -1.)

3. BILINEAR ALGORITHMS FOR MM

We need the next auxiliary result from the algebraic complexity theory (see [6, 16, 17]).

THEOREM 3.1.

If $O(n^s)$, $s<3$, arithmetic operations are involved in a straight-line algorithm for $n\times n$ matrix multiplication, then for arbitrary $\epsilon>0$ there exist integers $N>1$ and $Q=Q(N)$ and real constants $\nu>0$, $a_{ij}^q=a_{ij}^q(N)$, $b_{ij}^q=b_{ij}^q(N)$, $c_{ij}^q=c_{ij}^q(N)$ for $i,j=0,1,\dots,N-1$, $q=1,\dots,Q$, such that

$$m_q=L_q^1L_q^2,\ L_q^1=\sum_{i,j}a_{ij}^qx_{ij},\ L_q^2=\sum_{i,j}b_{ij}^qy_{ij},\ q=1,2,\dots,Q, \tag{3.1}$$

$$z_{ij}=\sum_{q=1}^{Q}c_{ij}^qm_q,\ z_{ij}=\sum_{k=0}^{N-1}x_{ik}y_{kj},\ i,j=0,1,\dots,N-1, \tag{3.2}$$

$$Q<N^{s+\epsilon},\ s<3, \tag{3.3}$$

$$f(N)=\max_{i,j,q}\{|a_{ij}^q|,\ |b_{ij}^q|,\ |c_{ij}^q|\}<N^\nu. \tag{3.4}$$

Here x_{ij}, y_{ij} are considered indeterminates and (3.1), (3.2) identities.

Obviously, if N is fixed, inequality (3.4) holds for all sufficiently large ν.

Notational Remark 3.1. Hereafter in some cases we need to refer to the constants without distinguishing among a_{ij}^q, b_{ij}^q, c_{ij}^q. In such cases we write f_{ij}^q where f stands for either a, or b, or c.

Hereafter let an integer $\bar{N}$ be fixed, and let $D(\bar{N})$, a decomposition (3.1), (3.2) for $N=\bar{N}$, and consequently $A(\bar{N})$, an algorithm generated by (3.1), (3.2) for $\bar{N}\times\bar{N}$ matrix multiplication, be fixed. Then using $A(\bar{N})$ recursively, derive $A(N)$, algorithms for $N\times N$ matrix multiplication for all N (see [1]). For this, assume that for all i, j the indeterminates, x_{ij}, y_{ij}, are replaced by $\bar{N}\times\bar{N}$ matrices. Then $X=(x_{ij})$, $Y=(y_{ij})$, $Z=(z_{ij})$, $Z=XY$, are $\bar{N}^2\times\bar{N}^2$ matrices, and all L_q^1, L_q^2 are $\bar{N}\times\bar{N}$ matrices. Apply $A(\bar{N})$ to multiply all pairs L_q^1, L_q^2 in the original algorithm, $A(\bar{N})$. This gives $A(\bar{N}^2)$, an algorithm for $\bar{N}^2\times\bar{N}^2$ matrix multiplication. Similarly, we obtain $A(\bar{N}^h)$, $h=3,4,5,\dots$, algorithms for $\bar{N}^h\times\bar{N}^h$ matrix multiplication. ($A(\bar{N}^h)$ is called the hth tensorial power of $A(\bar{N})$.) Applying $A(\bar{N}^h)$ to $\bar{N}^h\times\bar{N}^h$ matrices X, Y banded with zeroes, we obtain $A(N)$, an algorithm for $N\times N$ matrix multiplication where $\bar{N}^{h-1}<N\le\bar{N}^h$.

For all N the algorithm $A(N)$ defines $D(N)$, a bilinear decomposition (3.1), (3.2), and consequently $f_{ij}^q(N)$, the constants of $D(N)$,; see Notational Remark 3.1. (Expand all linear expressions of x_{ij}, y_{ij} contained in L_q^1, L_q^2.) Then the following properties can be easily inspected.

LEMMA 3.1.

For all N the constants of the algorithms $A(N)$ coincide with the constants of the algorithm $A(\bar{N})$ and of the decomposition $D(\bar{N})$. If $\bar{N}^{h-1}<N\le\bar{N}^h$, then

$$f_{ij}^q(N)=\prod_{l=1}^h f_{i(l)j(l)}^{q(l)}(\bar{N}), \tag{3.5}$$

where $i(l)$, $j(l)$, $q(l)$ are integers between 0 and $\bar{N}-1$ for $l=1,2,\dots,h$, and $f_{i(l)j(l)}^{q(l)}(\bar{N})$ are the constants of $A(\bar{N})$, $A(N)$, and $D(\bar{N})$.

LEMMA 3.2.

If relation (3.4) holds for $N=\bar{N}$, $\nu=\bar{\nu}$, then it also holds for all N and for all $\nu>\nu(\bar{\nu})$ where $\nu(\bar{\nu})$ is a constant independent of N.

Proof. Use (3.5). □

LEMMA 3.3

An algorithm $A(N)$ uses no divisions, no subtractions, and $T(N)$ additions and multiplications where $T(N)=O(N^{s+\epsilon})$ and s, ϵ are as in Theorem 3.1.

For the proof, use (3.3), and see [1]. □

4. TRUNCATION OF INPUTS OF $A(N)$. PRECISION OF APPROXIMATION

Represent the values of all inputs of $A(N)$, that is, of all variables x_{ij}, y_{ij} and constants $f_{ij}^q(\bar{N})$, as binary numbers with fixed point. Truncate them to an appropriate number of digits to be defined in the next paragraph and substitute the truncated values of inputs of $A(N)$ for their original values. Such a substitution transforms $A(N)$ into a new algorithm which will be denoted $A^*(N)$. Let w^* designate the truncated value of w for $w=x_{ij}$, $w=y_{ij}$, and $w=f_{ij}^q(\bar{N})$ for all i, j, q. Let the values $f_{ij}^{q*}=f_{ij}^{q*}(N)$ be defined by (3.5) where $f_{ij}^{q*}(N)$, $f_{ij}^{q*}(\bar{N})$ substitute for $f_{ij}^q(N)$, $f_{ij}^q(\bar{N})$, respectively. Similarly, let L_q^{1*}, L_q^{2*}, m_q^*, z_{ij}^* be defined by (3.1), (3.2) (first equation). $Z^*=(z_{ij}^*)$ may differ from X^*Y^*, where $X^*=(x_{ij}^*)$, $Y^*=(y_{ij}^*)$, and $f_{ij}^{q*}=f_{ij}^{q*}(N)$ may not coincide with any truncation of $f_{ij}^q=f_{ij}^q(N)$ if $N>\bar{N}$.)

Hereafter we assume that x^*_{ij}, y^*_{ij}, $f^{q*}_{ij}(\bar{N})$ are the values of x_{ij}, y_{ij}, $f^q_{ij}(\bar{N})$ truncated to the minimum number of digits such that the next relations hold.

$$\|X^* - X\| < E^*, \ \|Y^* - Y\| < E^*, \ E^* = E/2MN. \tag{4.1}$$

$$\delta^* = \max_{i,j,q} |f^{q*}_{ij}(N) - f^q_{ij}(N)|, \ 3\delta^* N^{7+2\nu} M^2 < E \quad \text{for} \quad \nu > \nu(\bar{\nu}). \tag{4.2}$$

Since the truncation cannot increase the absolute value, it follows that

$$\|X^*\| \le \|X\| < M, \ \|Y^*\| \le \|Y\| < M, \tag{4.3}$$

$$\forall q \forall i \forall j: \ |f^{q*}_{ij}(N)| \le |f^q_{ij}(N)| \le f(N). \tag{4.4}$$

For (4.3) apply also (2.2); for (4.4) use also (3.4), (3.5).

It follows from (2.1), (2.2), (4.1) that

$$\|X^* Y^* - XY\| \le \|X^*(Y^* - Y)\| + \|(X^* - X)Y\| < 2NME^* = E. \tag{4.5}$$

The next estimates follow from (3.1)–(3.5), (4.2)–(4.4).

$$\|Z^* - X^* Y^*\| \le F^* = \max_{u,v} \sum_{q,i,j,k,l} |a^{q*}_{ij} b^{q*}_{kl} c^{q*}_{uv} - a^q_{ij} b^q_{kl} c^q_{uv}| \cdot |x^*_{ij} y^*_{kl}|. \tag{4.6}$$

$$F^* \le 3N^4 QM^2 (f(N))^2 \delta^* < E. \tag{4.7}$$

IN (4.6) we write f^{q*}_{ij}, f^q_{ij} for $f^{q*}_{ij}(N)$, $f^q_{ij}(N)$. (Here f stands for a, b, c; see Remark 3.1.) Applying (4.5)–(4.7), we prove that the desired precision of approximation has been attained in the algorithm $A^*(N)$, that is,

$$\|Z^* - Z\| \le \|Z^* - X^* Y^*\| + \|X^* Y^* - Z\| < 2E.$$

5. ESTIMATES FOR THE NUMBER OF BITS AND BIT-OPERATIONS

In the previous section we obtained an algorithm $A^*(N)$ that evaluates $Z = XY$ with the required precision, $2E$, if it is applied to X^*, Y^*. Let us assume temporarily, until we have proved estimate (5.1) below, that all inputs and all constants of $A^*(N)$ have been replaced by their absolute values. This defines a new algorithm. Let μ designate the maximum among all its inputs, outputs, and intermediate results. Then the following estimates can be obtained using the relations (3.1)–(3.5) where f^q_{ij}, x_{ij}, y_{ij} are replaced by $|f^{q*}_{ij}|$, $|x^*_{ij}|$, $|y^*_{ij}|$ respectively.

$$\mu \le 4N^4 QM^2 (f(N))^3 < 4N^{7+3\nu} M^2, \ 2 + \log \mu = O(\log(NM)). \tag{5.1}$$

As is obvious, μ and $2 + \log \mu$ are upper bounds on the absolute values of all ingredients of $A^*(N)$, that is, of its inputs, intermediate results, and outputs, and respectively on the number of binary digits standing to the left of the radix point in the binary representation of those ingredients of $A^*(N)$.

What can be said about the bits that stand to the right of the point? We recall that their number is minimized under (4.1), (4.2).

(4.1) is satisfied if

$$d_0 = \min_{i,j}(d(x^*_{ij}), d(y^*_{ij})) > \log(2MN/E). \tag{5.2}$$

Here and hereafter $d(w)$ designates the number of bits standing to the right of the radix point in the binary representation of w.

We will use the following simple lemma.

LEMMA 5.1

Given $2H+2$ numbers $g_1, g_1^*, \ldots, g_H, g_H^*, U, E$, such that $\forall i$: $|g_i| < U$, $|g_i^*| < U$, $|g_i - g_i^*| < E$, then

$$|D(H)| \le HEU^{H-1},$$

where $D(s) = F^*(s) - F(s)$, $F^*(s) = \prod_{i=1}^{s} g_i^*$, $F(s) = \prod_{i=1}^{s} g_i$.

Proof. (by induction in s).

$$F^*(s+1) - F(s+1) = F^*(s)(g_{s+1}^* - g_{s+1}) + g_{s+1}D(s).$$

Hence

$$|D(s+1)| \le U^s E + U \cdot |D(s)|. \qquad \square$$

Applying (3.5) and Lemma 5.1, we obtain that (4.2) is satisfied if

$$d_1 = \min_{i,j,q}(d(a_{ij}^{q*}(\bar{N})),\ d(b_{ij}^{q*}(\bar{N})),\ d(c_{ij}^{q*}(\bar{N}))) > \log(h\bar{N}^{h\nu}/\delta^*). \tag{5.3}$$

Now we notice (compare (3.1), (3.2), (3.5)) that at most $2d_0 + 3hd_1$ digits are required to represent the fractional part of an operand of algorithm $A^*(N)$. Consequently (see (5.1)), all operands have totally at most

$$2 + 2d_0 + 3hd_1 + \log\mu = O(\log(NM/E)) \tag{5.4}$$

bits in their "fixed point" binary representation. (Recall that $h \le \log N/\log \bar{N} = O(\log N)$.)

Now Theorem 2.1 follows from the latter estimate (5.4), combined with Theorem 3.1, Lemma 3.3, and the known upper bounds on the number of bit-operations required to add and multiply two p-bit integers, that is, $O(p)$ for an addition and $O(p \log p \log\log p)$ for a multiplication; see [18]. $\square$

6. THE CASE OF RATIONAL CONSTANTS

Suppose that in Theorem 3.1 the basic straight-line algorithm for MM generated decomposition (3.1), (3.2) where all constants a_{ij}^q, b_{ij}^q, c_{ij}^q are rational (and this is the case for all existent fast algorithms for MM); then Theorem 2.1 can be proved in a simpler way. Namely, in such a case multiply all a_{ij}^q, all b_{ij}^q, and all c_{ij}^q by their common denominators, r_1, r_2, r_3 respectively. This turns all the constants into integers. Substitute those integers for the original rational constants, and thus define an algorithm for the evaluation of rXY using only integer constants where $r = r_1r_2r_3$. Apply such an algorithm to the evaluation of rX^*Y^* where X^*, Y^* satisfy (4.1). Then evaluate the quotients $(rX^*Y^*)/r$ with the precision E. In this case our estimates are simplified because we do not need anymore to truncate the binary numbers representing the constants. Similarly all input-variables can be turned into integers. Then we apply modular arithmetic followed by the final N^2 divisions. This seems to be a very convenient general method for applications of the stabilization of arithmetic algorithms.

7. APPLICATION TO THE SHORTEST PATH PROBLEM

The shortest path problem. Given the matrix $D = (d_{ij})$ of costs of edges of a graph with N nodes, then evaluate the costs of the shortest paths connecting all pairs of nodes.

We assume that $d_{ij} = +\infty$ if there is no edge between the nodes i and j and that otherwise d_{ij} are nonnegative integers. (If d_{ij} are nonnegative real numbers, then by rounding-off and scaling them, we come to the integer case again, although the scaling may lead to the increase of d_{ij}).

We also consider the following variation of the Shortest Path problem. *The Bounded Shortest Path Problem.* Given nonnegative $H(N)$, then under the conditions of the Shortest Path Problem evaluate all those costs of the shortest paths connecting the pairs of nodes that are bounded by $H(N)$.

Of course, for sufficiently large $H(N)$ this is again the Shortest Path Problem.

Applying the method described in [19], p. 206, we can reduce the Bounded Shortest Path Problem to $O(\log N)$ many successive solutions of the following problem.

Problem 7.1. Given $U=(u_{ij})$, $V=(v_{ij})$, two $N\times N$ matrices whose entries are either $+\infty$ or nonnegative integers bounded by $H=H(N)$,

$$\forall i \forall j: u_{ij}\leq H(N) \text{ if } u_{ij}\neq\infty,\ v_{ij}\leq H(N) \text{ if } v_{ij}\neq\infty,\ u_{ii}=v_{ii}=0, \tag{7.1}$$

then evaluate the matrix $W=(w_{ij})$ and $W(H)=(w_{ij}(H))$ where

$$w_{ij}(H)=w_{ij} \text{ if } w_{ij}<H,\ w_{ij}(H)=H \text{ otherwise},\ w_{ij}=\min_k(u_{ik}+v_{kj}),\ i,j=0,1,\ldots,N-1. \tag{7.2}$$

The latter problem can be solved using the following algorithm; see [14, 20].

ALGORITHM 7.1

Step 1. Compute two $N\times N$ matrices, $X=(x_{ij})$, $Y=(y_{ij})$ where

$$x_{ij}=2^{-mu_{ij}},\ x_{ij}=0 \text{ if } u_{ij}=\infty,\ y_{ij}=2^{-mv_{ij}},\ y_{ij}=0 \text{ if } v_{ij}=\infty, \tag{7.3}$$

$$m=\lceil\log(4N+3)\rceil. \tag{7.4}$$

Here and hereafter $\lceil x\rceil$ denotes the minimum integer that is not less than x.

Step 2. Compute $Z^*=(z_{ij}^*)$, the matrix of approximations with the precision $2^{-mh(N)-1}$ to z_{ij}, the entries of the matrix $Z=XY$, such that $z_{ij}^*=z_{ij}=0$ if $\forall k: u_{ik}+v_{kl}=\infty$. Here (see (7.1))

$$h(N)=\max_{i,j}(w_{ij} \text{ for } w_{ij}\neq\infty),\ h(N)\leq H(N). \tag{7.5}$$

In order to find all $z_{ij}^*=0$ substitute ones for all nonzero x_{ij}, y_{ij} and multiply the resulting matrices. The zero entries of the product define $z_{ij}^*=0$.

Step 3. Compute the output matrix, $W^*=(w_{ij}^*)$, from $Z^*=(z_{ij}^*)$, using the following relations.

$$w_{ij}^*=\infty \text{ if } z_{ij}^*=0. \tag{7.6}$$

$$w_{ij}^*=0 \text{ if } z_{ij}^*\geq 1/2. \tag{7.7}$$

$$w_{ij}^*=\lceil m^{-1}\gamma(2z_{ij}^*)\rceil \text{ if } 0<z_{ij}<1/2 \tag{7.8}$$

where $\gamma(t)$ denotes the number of initial zeroes in the fraction of t assuming that t is a "fixed point" binary number. □

Before the final Step 4 let us verify that the matrix W^* defined by the algorithm coincides with the desired W. Indeed, if $w_{ij}\neq\infty$, then $w_{ij}\leq h(N)$, $z_{ij}\geq 2^{-mh(N)}$, and hence $z_{ij}^*>0$. Therefore, $w_{ij}=w_{ij}^*=\infty$ if $z_{ij}^*=0$, see (7.6). Since

$$\Big|z_{ij}^*-\sum_{k=1}^{N}2^{-m(u_{ik}+v_{kj})}\Big|\leq 2^{-mh(N)-1},$$

it follows that

$$2^{-mw_{ij}-1}\leq z_{ij}^*\leq N2^{-mw_{ij}}+2^{-mh(N)-1}.$$

If $z_{ij}^*\neq 0$, (7.5) and the latter inequalities imply that

$$2^{-mw_{ij}-1}\leq z_{ij}^*\leq (N+1/2)2^{-mw_{ij}}.$$

Hence

$$w_{ij} - m^{-1}\log(2N+1) \le -m^{-1}\log(2z^*_{ij}) \le w_{ij}. \tag{7.9}$$

It follows that $w_{ij}=0$ if $z^*_{ij} \ge 1/2$. Hence $w_{ij} = w^*_{ij}$ if $z^*_i \ge 1/2$, see (7.7). On the other hand,

$$\gamma(t) \le -\log t \le \gamma(t)+1 \text{ if } 0<t<1.$$

It follows that

$$-\log(2z^*_{ij}) - 1 \le \gamma(2z^*_{ij}) \le -\log(2z^*_{ij}). \tag{7.10}$$

Combining (7.9), (7.10), we obtain that

$$w_{ij} - (\log(4N+2))/m \le \gamma(2z^*_{ij})/m \le w_{ij}. \tag{7.11}$$

(7.4) and (7.11) together give that

$$w_{ij} - 1 < \gamma(2z^*_{ij})/m \le w_{ij}.$$

Since w_{ij} is an integer, the latter inequalities and (7.8) imply that $w_{ij} = w^*_{ij}$ if $0 < z_{ij} < 1./2$. □

Now, naturally,

Step 4. Write $W = W^*$. Evaluate $W(H)$ using (7.2).
As follows from (7.2), (7.3), (7.5),

$$\gamma(2z^*_{ij}) \le mh(N). \tag{7.12}$$

Now we obtain the following upper bounds on the number of bit-operations involved in Steps 1–3.

Step 1. $O(N^2 \log N \log H(N))$. (It is essentially required to evaluate mu_{ij}, mv_{ij} for all i, j. The straightforward algorithm gives the estimate above; see (7.3), (7.4).)

Step 2. $O(N^{s+\epsilon}H(N)\log H(N)\log\log H(N))$ where s, ϵ are as in Theorem 2.1. (The estimate follows from Theorem 2.1 and relations (7.3)–(7.5).)

Step 3. The estimate $O(N^2 \log H(N)(\log\log N)^2)$ is defined by the bit-operation complexity of N^2 divisions of $\gamma(z_{ij})$ by m for all i, j which are to be performed with the precision 0.5; see (7.5), (7.6)–(7.8), (7.12). It is increased to $\sum_{i,j} \gamma(2z^*_{ij}) \le N^2 mh(N) = O(N^2H(N)\log N)$, see (7.4), (7.5), (7.12), if we add $\gamma(2z^*_{ij})$ bit-operations for the evaluation of each $\gamma(2z^*_{ij})$.

Step 4. $O(N^2 \log H(N))$.
Summarizing, we obtain the following upper estimate.

THEOREM 7.1
The Bounded (by given $H = H(N)$) Shortest Path Problem for a graph with N nodes and with nonnegative integer costs of edges can be solved involving $O(N^{s+\epsilon}H)$ bit-operations where H is the maximum cost of an edge of the graph, and s, ϵ are as in Theorem 2.1 (see also Remark 2.1).

Remark 7.1. We ignore the multiple $f(H) = \log H \log\log H$ in the bound of Step 2 assuming that $f(H) = O(N^\epsilon)$ for all $\epsilon > 0$. (Otherwise, $O(N^sH)$ exceeds even the obvious upper bounds, $O(N^3 \log H)$.) Similarly we ignore the multiple $O(\log N)$.

Remark 7.2. If it is known that all shortest distances are $O(N^{0.5})$, then Theorem 7.1 gives the improvement over the upper bound of [14]. Of course, Theorem 7.1 is of no value if the shortest distances to be evaluated can be as great as, say, $N/4$.

Acknowledgements—I wish to thank Prof. P. Bloniarz, SUNY, Albany, and Prof. A. Meyer and R. Rivest, M.I.T., for discussions on the Shortest Path Problem, and Prof. F. Romani, Pisa, for sending me his manuscript[20]. I am also grateful to Evelyn Laurent for typing my paper.

REFERENCES

1. V. Strassen, Gaussian elimination is not optimal. *Numer. Math.* **13**, 354–356 (1969).
2. V. Ya. Pan, Strassen's algorithm is not optimal. *Proc. 19th Ann. Symp. on Foundations of Computer Sci.* 166–176 (1978).
3. D. Bini, M. Capovani, G. Lotti, F. Romani, $O(n^{2.7799})$ complexity for matrix multiplication. *Inf. Proc. Letters* **8**(5), 234–235 (1979).
4. V. Ya. Pan, Field extension and trilinear aggregating, uniting and canceling for the acceleration of matrix multiplication. *Proc. 20th Ann. Symp. on Foundations of Computer Sci.* 28–38 (1979).
5. V. Ya. Pan, New fast algorithms for matrix operations. *SIAM J. Comput.* **9**(2), 321–342 (1980).
6. V. Yal Pan, New combination of methods for the acceleration of matrix multiplication. *Comput. Math. Applics.* **7**, 73–128 (1981).
7. A. Schönhage, Total and partial matrix multiplication. *SIAM J. Comput.* to be published.
8. F. Romani, Some properties of disjoint sums of tensors related to matrix multiplication. *Nota Interna B80-4, IEI, Pisa,* Italy (1980).
9. D. Coppersmith and S. Winograd, to appear.
10. W. Miller, Computational complexity and numerical stability, *Proc. Sixth Ann. ACM Symp. on Theory of Computing.* 317–323 (1974).
11. V. Ya. Pan, The bit-complexity of arithmetic algorithms, *J. Algorithms* to be published.
12. A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems.* American Elsevier, New York (1975).
13. G. Yuval, An algorithm for finding all shortest paths using $N^{2.81}$ infinite precision multiplications. *Inf. Proc. Letters* **4**(6), 155–156 (1976).
14. M. L. Fredman, New bounds on the complexity of shortest path problem. *SIAM J. Comput.* **5**, 83–89 (1976).
15. F. Romani, Complexity Measures for Matrix Multiplication Algorithms. *Calcolo* **17**(1), 77–86 (1980).
16. V. Ya. Pan, On schemes for the computation of products and inverses of matrices (in Russian). *Russ. Math. Surveys,* **27**(5), 249–250 (1972).
17. V. Strassen, Evaluation of rational functions. In *Complexity of Computer Computations* (Edited by R. E. Miller and J. W. Thatcher), pp. 1–10. Plenum Press, New York (1972).
18. A. Schönhage and V. Strassen, Schnelle Multiplikation Grosser Zahlen. *Computing* **7**, 281–292 (1971).
19. A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Addison-Wesley, New York (1974).
20. E. Romani, Shortest Path problem is not harder than matrix multiplication. *Inf. Proc. Letters* **11**(3), 134–136 (1980).